Published in Achieving Software Product Quality, Erik van Veenendaal, Julie McMullan (eds.),
UTN publishers, 1997, Den Bosch, The Netherlands

A Test Management approach for structured testing

Erik van Veenendaal and Martin Pol

*Despite encouraging results with various quality improvement approaches, the IT industry is still far from achieving zero defect software. Testing will remain an important activity within software development and maintenance, often taking more than 30 - 40% of the total budget. Both the increasing importance of software in the society and the costs that are involved in testing, confirm the need for structuring the testing process. This paper provides an outline description of TMap, the **T**est **M**anagement **ap**proach for structured testing (both white-box and black-box) of software products. It provides answers to the what, when, how, where and who questions of testing. To structure the organisation and execute the test processes TMap is based on four cornerstones:*

- *a development process related life cycle model for the testing activities (L);*
- *solid organisational embedding (O);*
- *the right resources and infrastructure (I);*
- *usable techniques for the various testing activities (T).*

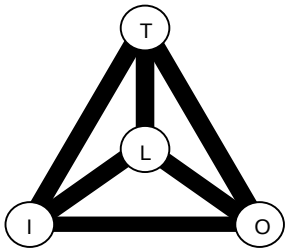


Figure 1 : the four cornerstone for structured testing

In recent years TMap has evolved towards the standard for software testing in The Netherlands. It is being used by more than two hundred Dutch organisations. Most Dutch banks, insurance companies, pensions funds and government departments use TMap partly or as a whole. More and more SME's have adopted TMap and new market segments have been penetrated such as consumer electronics, telecommunications and logistics. The TMap book (in Dutch) has proven to be a best seller, international interest and awareness has resulted in a plan for the release of an English version.

TESTING AS A PROCESS

For all types of testing the main activities are planning, preparation and execution. An important part of the effort is spent during planning and preparation. As a rule of thumb the following effort distribution can be used, 20% planning, 40% preparation and finally "only" 40% during test execution. Without going into great detail, this means that during the creation of the functional specification a master test plan is established. The master test plan describes who performs which type of testing and when. Ideally this master test plan covers all types of tests, from unit testing to acceptance testing. However, sometimes the scope is limited to only black-box testing (system and acceptance testing) or only to development testing (white-box testing and system testing). Since these types of tests plans affect several disciplines, the various objectives, tasks, responsibilities and deliverables have to be described accurately.

Improve Quality Services, Waalreseweg 17, 5554 HA Valkenswaard, The Netherlands
Tel 040 – 2089283 • Fax 040 – 2021450 • E-mail info@improveqs.nl

In larger projects an assignment with respect to the creation and subsequent co-ordination of the execution of such a test plan is often the responsibility of the manager of an independent (black-box) test team. On the basis of an agreed master test plan, more detailed test plans are made, mostly one for white-box testing, one for system testing and one for acceptance testing. These separate test plans are the responsibility of the various parties that are involved in the testing process.

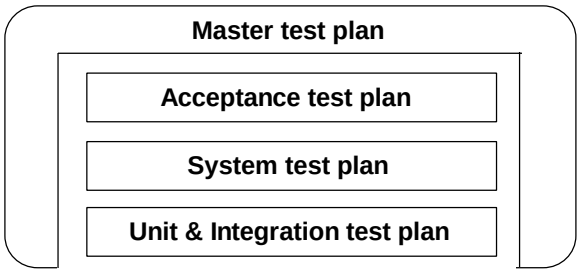
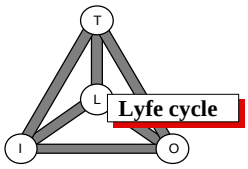


Figure 2: Hierarchy of test plans

After defining the test plans, in parallel to specification, design and coding activities, the test cases and the test infrastructure can be developed. After the delivery of the test object, the test cases will be executed. Structured testing introduces in addition to system design a second design process: test design. It would appear to be an expensive activity, but with careful planning, thorough risk-taxation, a well-founded testing strategy and an early start, costs are reduced considerably. Practice shows that the design of test cases, including the review of the requirements specification reveals a large number of defects, thus paying back the costs before the first tests have even been executed. It is known that rework effort on defects increases exponentially per development phase (Boehm,1979).

TEST LIFE CYCLE MODEL



The testing activities can be organised by means of a life cycle model, that operates in parallel with the life cycle models for system development. In the TMap®[1] life cycle model the three main testing activities are divided into five phases. In addition to the planning and control, preparation, specification and test execution phase, a completion phase has been defined to finish the testing process in a structured way and to preserve the testware for the first or next maintenance release.

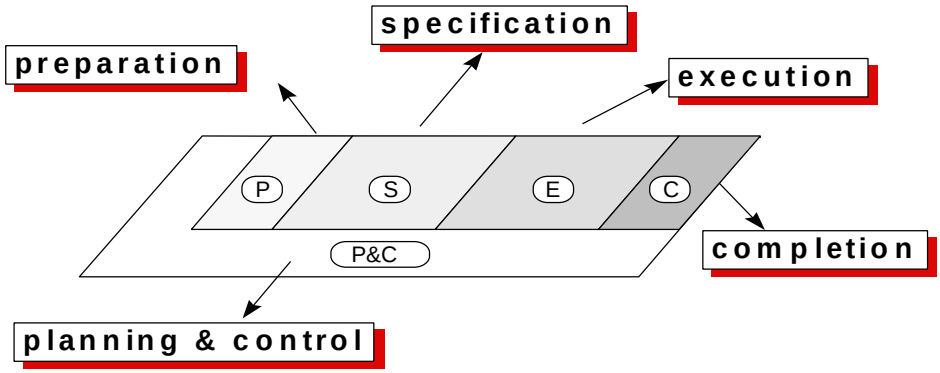


Figure 3: The TMap life cycle model

[1] TMap is a registered trademark of IP/Informatica Projectgroep B.V., The Netherlands

Improve Quality Services, Waalreseweg 17, 5554 HA Valkenswaard, The Netherlands
Tel 040 – 2089283 • Fax 040 – 2021450 • E-mail info@improveqs.nl

The TMap life cycle model is a generic model. It is, in principle, applicable to all types of testing. For white-box testing it contains too many activities. Only in highly critical circumstances will all activities be applicable. It is to up to the test manager to select the required elements from the possibilities offered by TMap. Within the hierarchy of the test plans several tailored TMap models will be operational for different types of tests. For the rest of this paper the outline of the generic model is described. This also applies to the terminology; "specification" should be interpreted as requirements specification *and* as technical design.

The planning and control phase

The "Planning and control" phase starts during the specification of the functional requirements. The planning phase provides the basis for a manageable and high-quality testing process. No matter how hard, at an early stage during system development all those things which make the testing process difficult to manage and control have to be discussed. For example the actual value of the development planning, the expected quality of the test object, the organisation of the various tasks and the availability of staff, infrastructure and time. The planning phase is the most important testing phase, however, it is often underestimated.

After the test assignment has been agreed, which is absolutely necessary, the test team starts by studying the specification, the functionality and the (project) organisation. It is impossible to test a software product completely. In theory, 100% coverage is possible, but no organisation has the time and money to achieve which is why the testing strategy is determined by means of risk analysis. Which parts of the system will get more attention during testing and which less etc. depending on the risks involved. Defining the test strategy is basically a communication process with all parties involved, trying to define the most important parts of the software product. The aim is to have the most feasible coverage on the right parts of the software product. In addition to this the first steps are taken towards structuring the testing organisation and defining the test infrastructure. All these activities are performed at the beginning of the testing process.

The other activities of the "Planning and control" phase are carried out throughout the entire testing process with the objective of managing the progress of testing with regard to the time and resources used. If necessary, detailed plans are drawn up for each phase during the testing process. In accordance with the test plan, reporting is done on the progress of testing and on the quality of the test object. The most important deliverable of testing is the quality report which also describes the accompanying risks. Right from the start of the testing process, testers are developing a view of the quality. It is important that during all phases of the testing process quality indicators are established. Periodically, and when asked ad hoc, management receives a quality report on the status of the test object. This is done continuously, not only at the end of the process. It is necessary that the test team provides well structured management information. One cannot just say just before the end of the testing process: No, you cannot go to production, I have not finished testing yet! No manager wants these types of comments. A manager wants to know the risks and what actions should be taken. It is, for example, possible to decide to continue testing, partly go in production or to keep the old system operational in parallel to the new one (shadowing).

Improve Quality Services, Waalreseweg 17, 5554 HA Valkenswaard, The Netherlands
Tel 040 – 2089283 • Fax 040 – 2021450 • E-mail info@improveqs.nl

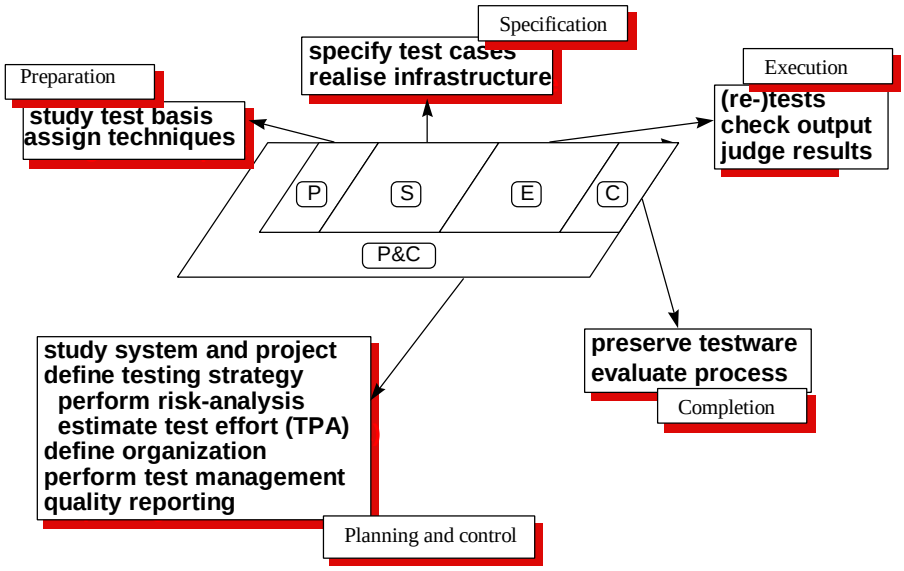


Figure 4: Activities within the TMap life cycle model

The preparation phase
The preparation phase starts as soon as possible after the test plan has been drawn up and agreed upon. The first activity during this phase is the training of test staff. Once the first version of the specification is ready, and has an "adequate" level of quality, e.g. most imperfections has been removed, the actual preparation activities can start. It most important that there is a stable version of the specification. The specification is the starting point for both testers and developers. After the establishment of the first version, the specification can only be changed by way of formal change control. In many cases establishing the first real version takes a long time. The aim is a 100% accurate specification, but this, unfortunately, can not be achieved. The temptation to start is strong but should be discouraged. Often test design, technical design and coding activities started prematurely have to be done again. That is frustrating and for all parties involved and very expensive.

The preparation phase starts with the detailed study of the specification and other documentation that serve as a starting point for testing. Insight is acquired on the testability of the specification by reviewing. During the review aspects like the use of standard notations, understandability and recognizability are important. Using the results of the review, the quality of the test basis, e.g. the specification, can be increased. After the study, and with the co-operation of the developers, the test basis will be divided into sub-systems that can independently be delivered and tested. Subsequently the test techniques are allocated to each of these test units and a plan is made for the next test phase and its activities.

The Specification phase
During the specification phase the test cases are specified and the accompanying test infrastructure is realised. The creation of test cases is carried out in two phases, the logical and the physical test design. Once the test basis are available, the logical test cases are specified (test specification). A test case consists of a description of the input, the process to be executed and a prediction of the expected output, e.g. results. Later, when more information is available about the technical implementation, the logical test cases are translated into physical test cases (test scripts). During this process the initial content of the test database is also defined. In parallel to the test design, the test infrastructure (the hardware- and software environment, etc.) is being set up.

The execution phase
The execution phase starts at the moment the first testable components of the software product are available. Agreements have been made with the development teams during previous phases about the

Improve Quality Services, Waalreseweg 17, 5554 HA Valkenswaard, The Netherlands
Tel 040 – 2089283 • Fax 040 – 2021450 • E-mail info@improveqs.nl

delivery schedule and the infrastructure to be used. First the delivered parts of the software product are checked for completeness and installed in the test environment. After this a test is carried out to establish whether the application environment and technical infrastructure can run without immediate failures. To be able to start the actual tests, the initial test database has to be set up, this is a very important and accurate activity and should be done by using the actual software functions as much as possible. In this way testing has really already started.

When (parts of) the software product, the infrastructure and the test database are available, the first so called pre-tests are executed to check whether the main functions of the object can be tested. The pre-tests provide an answer to the question: Is the quality of the test object such that it can efficiently and effectively be tested using the prepared test cases? As soon as the pre-tests have been completed successfully, test execution can start using the test scripts. The execution takes place on the basis of the agreed test strategy. The difference between the actual test result and the expected result can indicate a software product defect, but can also indicate a defect in the specification, a defect in the test infrastructure or an invalid test case. The cause of the observed difference will be further investigated during the checking and judgement activities. As soon as rework has been completed, the tests are executed again.

During the entire test execution phase one should allow for quick and reliable quality reporting. Management will expect to be informed about the risks they must consider. They want to know: What percentage of the product has been tested?, What remains to be done?, How many defects have been found?, What are the trends?, Can testing be finished?, etc.

The completion phase

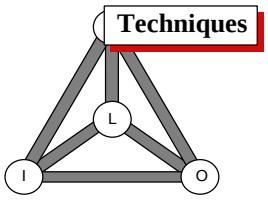
When test execution has been finished, there are still some important activities to be done. These activities are generally carried out in a less structured way, or even forgotten. The final test execution activities often take place under high pressure. This will mean concessions are made to the control procedures. The test scene is often chaotic at the time testing “ends” and the decision is made to go into production. An additional problem that often arises is the phenomenon that during production new defects are found by the users and these must be solved and tested without delay. As a result the completion activities will often get low priority, and no time and effort is scheduled, the testware is not preserved in an adequate way and therefore can not be reused during the first or next maintenance test.

During the completion phase a selection is being made with respect to the large amount of testware, e.g. test cases, test results and descriptions of the test infrastructure and the tools used. This has to be done from the perspective of required product changes, the related maintenance tests will only need to be adjusted and no completely new test scripts will have to be designed. During the testing process an effort is made to keep the test cases consistent with the specification and the software product. When this is carried out successfully, one can truly speak of so-called regressive testware. Keeping the consistency between testware, specification and the actual software product is an important objective during the maintenance.

During the completion phase the testing process is being evaluated. The statistics gathered and intermediate reports are combined with the results of the final report. Both the testing process and the quality of the product are being evaluated. It is recommended that an overview of costs and benefits of the testing process is drawn up, a difficult but also very engaging and necessary activity. The often large, quantities of statistics that are available are essential to improve future planning and optimisation of the testing processes, development processes and the quality system. After the evaluation, the preservation of the testware and the presentation of the final evaluation report, management will be able to discharge the test team.

Improve Quality Services, Waalreseweg 17, 5554 HA Valkenswaard, The Netherlands
Tel 040 – 2089283 • Fax 040 – 2021450 • E-mail info@improveqs.nl

TECHNIQUES



TMap is supported by a large number of testing techniques. These techniques provide test staff with a well-founded and similar way of working and management (and auditors) with the possibility of tracking the testing process with respect to the contents. Throughout the TMap life cycle model the tester is directed towards the techniques that can be used. Figure 5 shows the testing techniques that are available within Tmap.

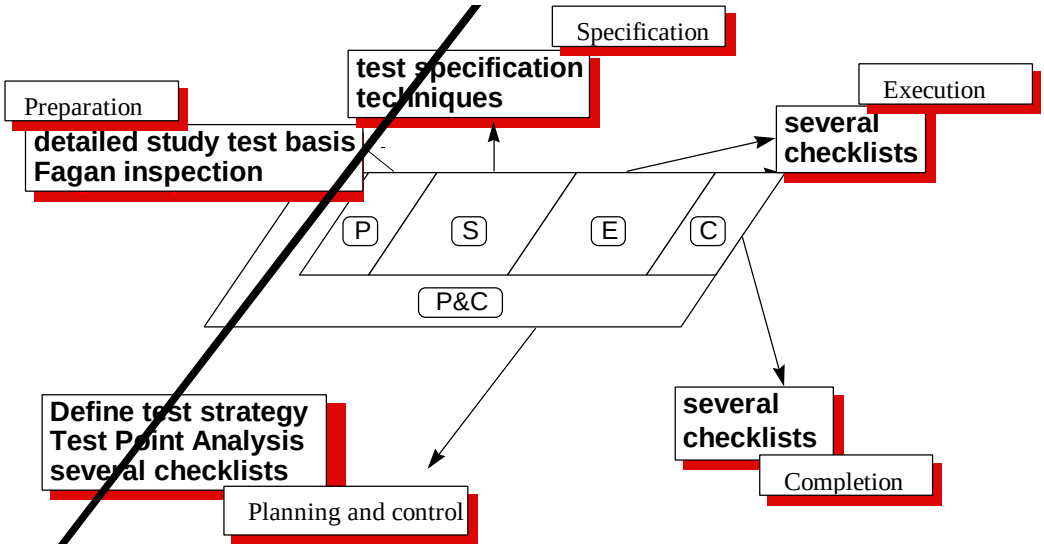


Figure 5: Testing techniques

Each of the identified testing techniques is briefly described in the following sections.

Defining testing strategy
The definition of the test strategy takes place during the planning phase of the testing process. During the planning phase the basis is laid for a manageable and high-quality testing process. Defining the test strategy is a mechanism for test management to communicate with the sponsor, users, developers and other parties involved, about the organisation and strategic choices of the testing process. The test strategy defines what is going to be tested and how thoroughly it is going to be tested. Choices have to be made, since it is impossible to test a software product completely; 100% coverage on all functionality and quality characteristics is perhaps possible in theory, but no organisation has the time and money to do it. A risk analysis will be the starting point for a well defined test strategy. The test strategy is further refined and determined via a process of communication by all parties involved, trying to define the most important parts and quality characteristics of the software product. The aim is to have the most feasible coverage on the right parts and quality characteristics of the software product.

Test Point Analysis
Once the test procedure has been determined, the accompanying testing effort can be estimated. How much time is needed to test sub-system X by means of testing technique Y by employee Z. To get an answer to these difficult questions metrics are needed. Testing is still relatively immature and there are still only few statistics and metrics available. Often the testing effort is determined as a percentage from the development effort. However it is becoming common practice to determine the size of a software product by using Function Point Analysis (FPA). On the same basis Test Point Analysis (Veenendaal,1995) has been developed and applied successfully to estimate the required testing effort. Test point analysis is suitable for estimating the testing effort for a black-box test, e.g. a system or acceptance test. Test point analysis converts function points to test points from which the number of

Improve Quality Services, Waalreseweg 17, 5554 HA Valkenswaard, The Netherlands
Tel 040 – 2089283 • Fax 040 – 2021450 • E-mail info@improveqs.nl

hours required for testing can be calculated. The technique analyses the impact of particular test influencing factors, such as the quality characteristics to be tested, number of interfaces, complexity of the software product, quality of the specification and test tools that can be used.

Detailed study of test basis
The test basis consists of the documentation on which the tests will be based, e.g. from which the test cases will be derived. It goes without saying that a in-depth study or review of the test basis is an important consideration to guarantee adequate progress during the actual preparation of the test. This provides is a review of the documentation for completeness, accuracy and consistency to judge whether it can serve as an starting point for testing. The detailed study is often carried out using the Fagan inspection technique (Fagan,1976).

Test specification techniques
Using the specification **test cases** must be determined. An elementary test case consists of a specification of the starting situation, the changing process and a prediction of the expected result.

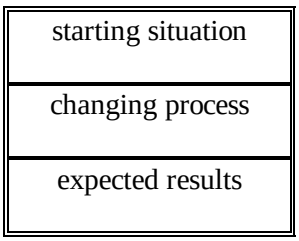


Figure 6: Elementary test case

A test case can aim at testing one or more quality characteristics of one or more features of the software product. *Test specification techniques* are available to deduce the test cases from the specification. The individual test cases are subsequently grouped into test scripts that prescribe which starting situation and which sequence of actions and checks are applicable during the execution of the test. The collection of test scripts are combined into an overall test procedure that also describes the relationship with required test infrastructure.

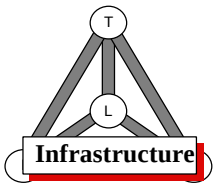
Test specification technique:
A test specification technique is a standardised way to derive test cases from the development documentation, e.g. requirements specification or technical design (Pol et al, 1995).

For testing the various quality characteristics different specification techniques are available. The way a technique is applied depends on the way the specification is structured. Taking the testing strategy and the structure of the development documentation as a starting point, the appropriate test specification techniques are selected and tailored during the preparation phase.

Checklists
TMap offers a large number and variety of checklists. For the planning and control and preparation phase. The checklists available can be used for support during the study and review, the definition of constraints and starting points, the risk analysis and the definition of the test facilities and infrastructure. Quality characteristic checklists are available to support the test execution phase, especially with respect to static tests. Evaluation checklists support the final reporting and completion activities.

INFRASTRUCTURE AND TOOLS

Improve Quality Services, Waalreseweg 17, 5554 HA Valkenswaard, The Netherlands
Tel 040 – 2089283 • Fax 040 – 2021450 • E-mail info@improveqs.nl



The infrastructure for testing includes all facilities and resources needed for structured testing. A distinction can be made regarding the test infrastructure between test environment, test tools and office environment. Choosing an infrastructure for testing is often not an option. A test environment is already available, there are already test tools and an office environment for testing exists. The choice is then limited to change requests regarding the existing infrastructure. In addition to this the infrastructure strongly depends on the type of the hardware platform and the organisation. Because of these issues TMap is limited to the generic level and can only offer some supporting ideas.

Testing environment

Traditionally three types of test environments are available. The laboratory environment for white-box testing, the (more) controllable system test environment and the "as-if-production" environment for acceptance testing (see Figure 7). Due, often to efficiency considerations, the particular circumstance of testers having their "own" environment is being abandoned. It is much more efficient to determine the testing environment needed on the basis of the type of test that has to be carried out.

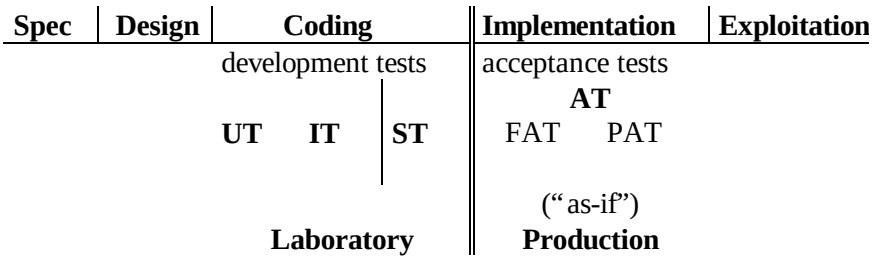


Figure 7: Traditional testing environments

White-box testing requires a completely different test environment than, for example, production-acceptance tests (PAT). However, sometimes parts of a production acceptance test do not have to be performed in an "as-if" production environment. Which quality characteristics are tested? Are formal procedures and/or actual database sizes necessary? Or is having the possibility to do a quick adjustment to the test object more important for the progress of the testing process? These types of questions have to be answered at an early stage of the testing process, especially since it takes time to prepare the facilities and environment. The computer centres, that usually offer these types of facilities to the testers, have to be informed of the requirements accurately and on time. To prevent disappointment, it is strongly recommended (or even necessary) to get assistance from computer centre staff during the definition of these requirements.

Test tools

The development of test tools has matured over the past years. The application and variety has increased enormously. As for CASE (Computer Aided Software Engineering), there are publications about CAST (Computer Aided Software Testing). CAST can be looked upon as an enumeration of the available test tools, classified by application and platform. Of special interest are test tool that are multi-platform, of even platform independent. Test tools can be subdivided according to the support they provide to the TMap life cycle activities. The most important support is currently being provided to the planning and control phase and test execution phase. For *planning* and *progress tracking* the same tools can be used as in every other process, e.g. planning packages, spreadsheets and risk-analysis packages. There is also an automated version available of the test estimation technique, test point analysis. For the *execution* and *judgement* of tests the following test tools can be used among others:

- *Record (or capture) & playback*

These tools record testing sessions that can later be "replayed" automatically;

- *Comparators*

Improve Quality Services, Waalreseweg 17, 5554 HA Valkenswaard, The Netherlands
Tel 040 – 2089283 • Fax 040 – 2021450 • E-mail info@improveqs.nl

An automatic comparison of test results with results of a previous testing session;

- *Test drivers*

A replacement of a driver program that has not yet been delivered, to allow testing of the underlying part of the software product;

- *Simulators*

Tools in which the "real-life" environment can be simulated to be able to test for, for example, the performance and resource-utilisation;

- *Coverage analysers*

With coverage analysers the extent to which the test object is covered by the test cases can be measured;

- *Static analysers*

A Static analyser can provide metrics on software to measure, for example, the maintainability and can identify non-conformances to a coding standard.

For the *remaining phases* tools are available which, for example, support the loading and control of the test databases, configuration management, the registration of test defects and the gathering and presentation of statistics and performance indicators.

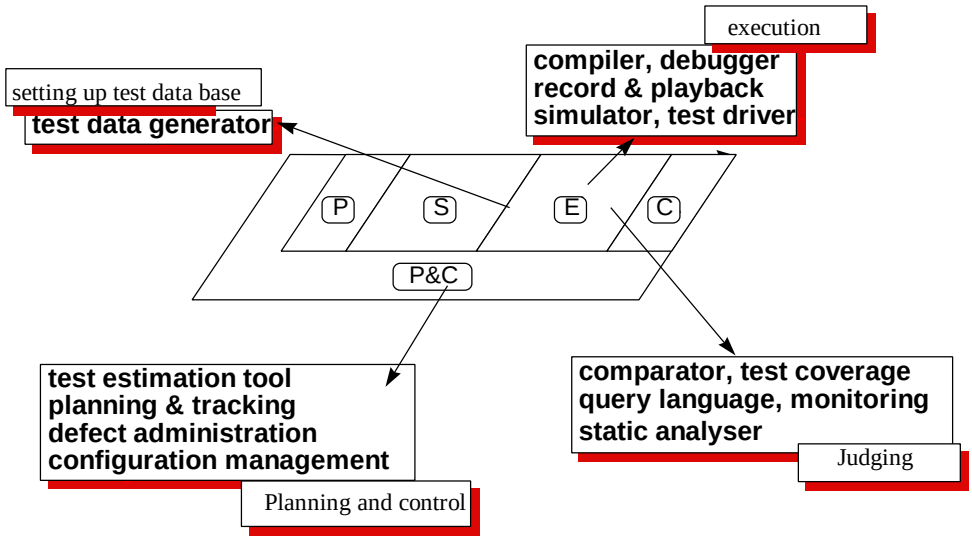


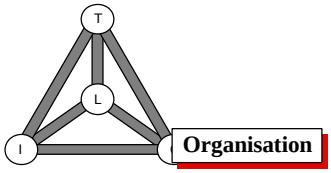
Figure 8: TMap and CAST

Office environment

The testing process, from planning to completion, demands a suitable office environment. This may seem self-evident, however practice shows that testers do not have desks and PC's available on time, or they have to share them with their colleagues. The test workstations have to be fully operational at the start of the specification phase, and not only at the beginning of the test execution phase. Quite often large cost savings can be realised by using the same PC's, during both test specification and test execution. Testing often requires a lot of consultation, so a dedicated meeting room would be useful. Also much testing is undertaken outside the regular office hours. Therefore catering provisions, security facilities and for example transport possibilities at difficult times should be taken into consideration. It is also recommended to reduce, as much as possible, the geographic distance between the computer centre, the developers and the testers.

Improve Quality Services, Waalreseweg 17, 5554 HA Valkenswaard, The Netherlands
Tel 040 – 2089283 • Fax 040 – 2021450 • E-mail info@improveqs.nl

ORGANISATION



Nearly every testing process with insufficient organisation will end in a failure. The involvement of many different disciplines, the unpredictability of the process, the complex control activities, a lack of experience and the time pressures put severe demands on the testing organising and the various control activities

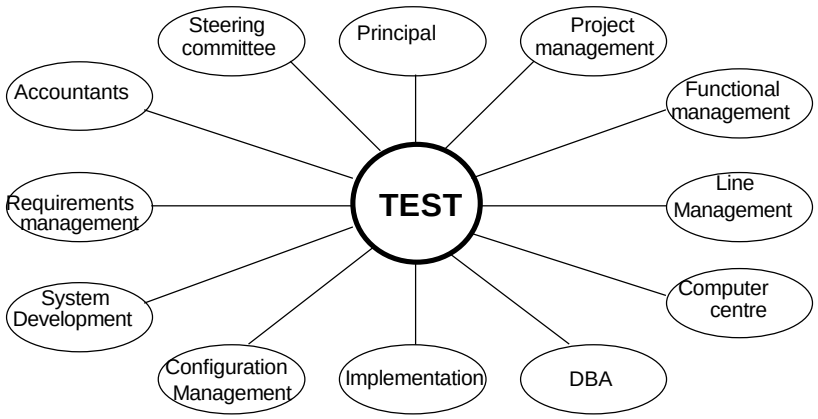


Figure 9: The many different disciplines involved

Testing organisation :
A testing organisation is the representation of the relationships between test functions, test facilities and testing activities aimed at performing a structured test (Thierry,1973).

The following aspects need attention in the context of organising structured testing:
- the operational testing process
- the structural testing organisation
- test management
- staffing and education
- structuring of the testing process.

The operational testing process
On a departmental or project level where the actual testing processes take place, it is important to set up a flexible, but at the same time stable, organisation. This calls for experience, tact, patience and the timing skills of (test) management. The establishment of testing functions and especially control resources is an on-going process of making choices between progress and quality. Getting staff or acquiring tools too early or too late are classical problem areas in the testing process.

An operational testing organisation consists of a variety of functions. Depending on the type of test and the size of the test object, these functions are performed by one or more persons. Sometimes a test team consists of one part-time employee, in large projects a (black-box) test team may involve twenty or more employees. The most important test functions are: testers, test management and test control. TMap recommends carrying out black-box tests in a project organisation (test team). It is important to get the right combination of expertise in an operational testing organisation:
- knowledge and skills in the area of testing
- functional knowledge with respect to the test object, e.g. software product
- knowledge and experience of test management and control
- knowledge and experience in the area of infrastructure and tools.

Improve Quality Services, Waalreseweg 17, 5554 HA Valkenswaard, The Netherlands
Tel 040 – 2089283 • Fax 040 – 2021450 • E-mail info@improveqs.nl

The structural testing organisation
The organisation and control of the structural testing depends on a large number of items, such as the maturity level and the size of the organisation. Of course the IT dependency of an organisation is also a great influence; is it possible to allow an error to occur or will even a minor incident reach the front page of the Wall Street journal. How important is time-to-market? Are human lives depending on the software product or are "only" personal careers damaged?

The test functions can be subdivided into those related to providing adequate conditions and those related to providing operational functions. Functions related to providing conditions, are for example test regulation, co-ordination of the testing processes, test management, method support and technical support. In large organisations in which the software products are closely related to the primary processes, as in banking and insurance companies and some Government agencies, these types of test functions are often established at several levels of the structural organisation. The operational tasks, e.g. test execution, are then carried out on a project level. In other organisations a mixture exists or only the separation between test regulation and test execution is present. Often regulation and auditing functions are positioned in a Quality Assurance and/or Methods & Techniques department. In other cases a special testing office or a test service department is operational. Sometimes only something is organised on departmental or project level. In short, as already stated: *A universal type of organisation does not exist for testing.*

Test management and control
Management, taken from *managing*, basically means keeping under control. This sentence reflects the test management and control function. One of the typical problems of a testing process is that everything, even the environment, is changing all the time. These changes have to be controlled so as to give a solid quality indication with respect to the test object. Within the testing processes three management and control aspects can be distinguished:

- management and control of the testing process
- management and control of the testing infrastructure
- management and control of the test deliverables.

Depending on the type of organisation, management is carried out completely or partially at the operational level or at the level of the structural organisation. Some management tasks, like management of the test object and/or test database, are essentially arranged outside of the test process. Within management of the testing process, it is of greater importance to have the right attitude than to follow a strict set of rules. Management can be looked upon as an environmental control on the testing process. One does not allow a test database or defect to lie around; that is just *not done!*

Staffing & education

Staffing
Testing has evolved in the past years into a true profession. The growing need for solid tests and the related developments in the area of testing require specialist knowledge and skills and the accompanying education. This does not mean that testing is restricted to specialised testers. A user, a product manager or a developer can or has to perform testing in addition to his or her primary tasks. A test team consists of a large variety of disciplines which, some part-time, contribute their specific expertise. Adequate participation of testing specialists is of course essential, both in the area of test management and in the area of testing techniques. Test staffing requires a lot of (test) management attention. Aspects as working with part-timers that have testing as a secondary function and the unpredictability of staffing levels because (development) schedules run late, should not be underestimated.

To have the required test staff at the right moment, it is necessary to have a good insight into the test

Improve Quality Services, Waalreseweg 17, 5554 HA Valkenswaard, The Netherlands
Tel 040 – 2089283 • Fax 040 – 2021450 • E-mail info@improveqs.nl

functions and tasks and related knowledge and skills. This insight is required to communicate with the employees and trainers, and for the possible selection procedures for candidate testers.

Education
The function requirements ask that testers have a great variety of knowledge and skills. An education programme for test staff naturally has to contain test specific components such as test management and testing techniques. In addition to this, it also has to provide a general knowledge of quality assurance and system development. Also social skills should be covered. There are a variety of options, but like for many other educational programmes, a combination of theory and practical application is strongly recommended. The best result are obtained by means of a brief theoretical introduction followed by adequate training-on-the-job.

Structuring the testing process
The change process towards a more structured testing organisation has its own approach and dynamics. It often occurs that test structuring processes fail. This is generally due to underestimating the organisational and also financial implications, both at the implementation and application level. Also the reduced level of attention, after the test approach has been applied successfully can be identified as a reason for failure.

A test structuring process often originates from an expensive failure or even a disaster. Management then decides "something has to be done". Somebody is assigned to test structuring and meanwhile a pilot project or a software product is selected, with which the new approach has to prove itself. As already stated this often goes wrong. A structuring process asks for a strategy and know-how on both change management and testing. After the investigation of the state-of-the-practice a step-wise approach has to be followed to achieve the desired results. In this, timing and tact play an important role.

REFERENCES

Boehm, B.W. (1979), *Software engineering economics*, Prentice-Hall, Englewood Cliffs, NJ

Fagan, M.E. (1976), Advances in software inspections, in: *IEEE Transactions on Software Engineering*, IBM Systems Journal, Vol. 15, No. 3

Hetzel, W.C. (1988), The complete guide to software testing, QFD Information Sciences, Wellesley, ISBN 0-89435-242-3

IEEE (1994), *IEEE Standards Collection*: *Software Engineering*, Institute of Electrical and Electronic Engineers, inc., ISBN 1-66937-442-X

Meyers, G.J. (1979), *The art of software testing*, Wiley-Interscience, New York, ISBN 0-471-04328-1

Pol, M., R.A.P. Teunissen and E.P.W.M. van Veenendaal (1995), *Testing according to TMap* (in Dutch), 's Hertogenbosch, The Netherlands, ISBN 90-72194-33-0

Pol, M., R.A.P. Teunissen and E.P.W.M. van Veenendaal (1996), *Structured testing: An introduction to TMap* (in Dutch), 's Hertogenbosch, The Netherlands, ISBN 90-72194-45-4

Thierry, H. (1973), *Organisation and management* (in Dutch), Senfert Kroese, Leiden, The Netherlands, ISBN 90-207-0445-1

Veenendaal, E.P.W.M. van (1995), Test Point Analysis: a method for estimating the testing effort (in Dutch), in: *Computable*, May 1995

Improve Quality Services, Waalreseweg 17, 5554 HA Valkenswaard, The Netherlands
Tel 040 – 2089283 • Fax 040 – 2021450 • E-mail info@improveqs.nl

BIBLIOGRAPHY

Drs. Erik P.W.M. van Veenendaal CISA has been working in the IT-industry since 1987 carrying out assignments in the field of quality assurance, project control, EDP-auditing and software testing. After having fulfilled a number of management positions at a leading Dutch IT quality consultancy company, he has been involved in a number of European projects within the area of Software Product Quality. Erik van Veenendaal is the founder and managing director of Improve Quality Services. Improve Quality Services provides services in the area of quality management, usability and testing. At the Eindhoven University of Technology, Faculty Technology Management, he is involved in lecturing and carrying out research on test management. He is a joint author of “Testing according to Tmap”.

Martin Pol is one of the founders of structured testing in The Netherlands. As a test manager at IP/Software Control Testing he has been involved in a great number and large variety of testing projects and implemented structured testing in several organisations. He is the co-author of two books on structured testing and is a frequent speaker at both national and international conferences. He is the chairman of the Dutch special interest group on software testing (TESTNET).

Improve Quality Services, Waalreseweg 17, 5554 HA Valkenswaard, The Netherlands
Tel 040 – 2089283 • Fax 040 – 2021450 • E-mail info@improveqs.nl